



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/728,097	12/05/2003	Hyun-kwon Chung	1793.1113	7294

49455 7590 08/02/2007  
STEIN, MCEWEN & BUI, LLP  
1400 EYE STREET, NW  
SUITE 300  
WASHINGTON, DC 20005

EXAMINER
----------

OSBERG, THUY THANH

ART UNIT	PAPER NUMBER
----------	--------------

2179

MAIL DATE	DELIVERY MODE
-----------	---------------

08/02/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

211

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/728,097	CHUNG ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Thuy Osberg	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 05 December 2003.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-14 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-14 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All    b) ☐ Some \* c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. This communication is responsive to the original application filed 12/05/2003.

Claims 1-14 are pending and have been examined. This action is **Non-Final**.

### *Claim Rejections - 35 USC § 102*

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

3. Claims 1-4 and 13-14 are rejected under 35 U.S.C. 102(e) as being anticipated by Stone et al. (US Patent 6,504,554), hereinafter "Stone".

As to independent claim 1, Stone teaches a method of executing markup document applet by a browser (fig. 5, labels 502, 504, 506; col. 7, lines 14-28; col. 8, lines 41-58), comprising:

receiving a request for executing an applet from the browser (fig. 6, label 610; col. 7, lines 42-43; col. 10, lines 45-47, by adding the applet to the browser);

determining whether the applet is a bound applet or an unbound applet (col. 7, lines 45-58; col. 10, lines 21-31);

loading the requested applet into a virtual machine (fig. 2, label 204, col. 5, lines 24-42; col. 10, lines 45-47, that "runtime environment" is a platform-specific environment that interprets or

Art Unit: 2179

compiles the applet code and performs the high-level instructions therein. These applets are executed via a runtime environment which is a virtual machine that renders dynamic HTML); and if the applet is an unbound applet (col. 7, lines 34-36, that applets are at the unbound state), immediately issuing predetermined commands to the virtual machine to first set the unbound applet into an initiate state and then into a start state, respectively (col. 7, lines 42-58; col. 8 lines 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate/start the applet by placing it in the "peer available state").

**As to dependent claim 2,** Stone further teaches:

if the unbound applet is completed, issuing predetermined commands to the virtual machine to set the unbound applet into a stop state and into a destroy state (fig. 7D, label 722), respectively (col. 7, lines 66-67, col. 8, lines 1-4, 54-56; 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser).

**As to dependent claim 3,** Stone further teaches:

if determined that the applet is a bound applet and determined by the browser that a markup document connected to the bound applet has a grammatically correct structure, issuing a predetermined command to the virtual machine to set the bound applet into an initiate state (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code based

Art Unit: 2179

on proper syntax; col. 10, lines 29-31, and further by placing it in the “peer available state” which is the initiate/start state).

**As to dependent claim 4,** Stone further teaches:

issuing a command to the virtual machine each time the markup document is displayed to set the bound applet into a start state while the markup document is being rendered by the browser (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate/start the applet; col. 10, lines 29-31, and further by placing it in the “peer available state” which is the initiate/start state);

if the markup document is unloaded by the browser, issuing a command to the virtual machine to set the bound applet into a stop state (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser);

and issuing a command to the virtual machine to set the applet into a destroy state to delete the bound applet (fig. 6, label 614, col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser).

**As to independent claim 13,** Stone teaches a method, comprising:

classifying tagged applets of a markup document (col. 8, lines 67, col. 9, lines 1-6);

Art Unit: 2179

and controlling different execution life cycles of the tagged applets according to the classifying (fig. 6; col. 7, lines 30-48; (col. 8, lines 67, col. 9, lines 1-6).

**As to dependent claim 14**, Stone further teaches classifying the tagged applets into bound and unbound applets (col. 8, lines 67, col. 9, lines 1-6), and wherein according to the controlling, execution life cycle of a bound applet depends on the markup document life (fig. 6; col. 7, lines 30-48; col. 8, lines 67, col. 9, lines 1-6), and the execution life cycle of an unbound applet is independent of the markup document life (fig. 6; col. 7, lines 30-48, that by placing it in the "peer available" state it is independent of the HTML document, but available in the browser if needed).

***Claim Rejections - 35 USC § 103***

**4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:**

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not

Art Unit: 2179

commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

**5. Claims 5-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stone in view of Engstrom et al. (US Pub 2005/0120305), hereinafter "Engstrom".**

**As to independent claim 5,** Stone teaches an apparatus (fig. 1; col. 3, lines 46-58) executing a markup document applet (Fig. 5, labels 502, 504, 506; col. 7, lines 14-28; col. 8, lines 41-58), comprising:

a virtual machine which executes an applet related to the markup document (fig. 2, col. 5, lines 24-42; col. 9, lines 36-37; col. 9, lines 36-37, that "runtime environment" is a platform-specific environment that interprets or compiles the applet code and performs the high-level instructions therein. These applets are executed via a runtime environment which is a virtual machine that renders dynamic HTML);

a browser which receives the markup document from the memory (fig. 3; col. 5, lines 46-50; col. 10, lines 6-9) and outputs information on an applet related to the markup document included in the markup document (col. 5, lines 50-58);

and an application manager (fig. 2, label 210, col. 6, lines 13-22) which receives the applet information from the browser (fig. 2, label 210, col. 6, lines 13-22), retrieves the applet from an external data source (fig. 2, label 210, col. 6, lines 13-22), controls the retrieved applet to be stored in the memory (fig. 2, label 210, col. 6, lines 13-22), receives a request for executing the stored applet (col. 7, lines 42-43, by adding the applet to the

Art Unit: 2179

browser), loads the stored applet into the virtual machine (fig. 2, col. 5, lines 24-26; col. 10, lines 45-47), determines whether the loaded applet is a bound applet or an unbound applet (col. 7, lines 45-58; col. 10, lines 21-24), and if the loaded applet is an unbound applet, immediately issues predetermined commands to the virtual machine to first set the loaded unbound applet into an initiate state and then into a start state, respectively (col. 7, lines 45-58; col. 8 lines 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate/start the applet by placing it in the "peer available state").

Stone does not teach a memory which stores an input markup document.

However, Engstrom teaches a memory which stores an input markup document (par [0048]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Stone by having a memory which stores an input markup document as taught by Engstrom in order to provide high capacity storage to host the markup document that provides the input to the system to provide the functionality of launching the applets.

**As to dependent claim 6**, Stone further teaches when the virtual machine informs the application manager that the unbound applet is completed, and the application manager issues a command to the virtual machine to set the unbound applet into a stop state (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code) and issues another command to the virtual machine to set the unbound applet into a destroy state to unload



Art Unit: 2179

the unbound applet from the virtual machine (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser).

**As to dependent claim 7**, Stone further teaches if the applet is a bound applet, the browser informs the application manager that the markup document connected to the bound applet has a grammatically correct structure, and the application manager issues a command to the virtual machine to set the bound applet into an initiate state (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate/start the applet by placing it in the “peer available state”; col. 10, lines 29-31; col. 10, lines 29-31).

**As to dependent claim 8**, Stone further teaches the application manager further: issues a command to the virtual machine to set the bound applet into a start state while the markup document is being rendered by the browser (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate/start the applet by placing it in the “peer available state”), issues another command to the virtual machine to set the bound applet into stop state, if the markup document is unloaded by the browser (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code), repeatedly issues the initiate and stop commands to the virtual machine to start and stop the bound applet upon

Art Unit: 2179

redisplaying the markup document by the browser (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser and then by reattaching in a "peer available" state), and issues a command to the virtual machine to set the bound applet into a destroy state to unload the bound applet from the virtual machine (col. 7, lines 66-67, col. 8, lines 1-4, 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to stop/destroy the applet by placing the element in the unbound state stops/destroys it from interacting within the browser).

**As to independent claim 9, Stone teaches:**

receiving a request for executing an applet from the browser (col. 7, lines 42-43, by adding the applet to the browser);

determining whether the requested applet is a bound applet or an unbound applet (col. 7, lines 45-58; col. 10, lines 21-24);

loading the requested applet into a virtual machine (fig. 2, col. 5, lines 24-26);

and if the requested applet is an unbound applet (col. 7, lines 34-36, that applets are at the unbound state), immediately issuing predetermined commands to the virtual machine to first set the requested loaded unbound applet into an initiate state and then into a start state, respectively (col. 7, lines 45-58; col. 8 lines 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate and/or start the applet).

Art Unit: 2179

Stone does not teach a computer-readable recording medium storing at least one program controlling an interactive contents reproduction apparatus to execute a markup applet according to a process.

However, Engstrom teaches a computer-readable recording medium (par [0047]) storing at least one program controlling an interactive contents reproduction apparatus to execute a markup applet according to a process (par [0048]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Stone by having a computer-readable recording medium storing at least one program controlling an interactive contents reproduction apparatus to execute a markup applet according to a process as taught by Engstrom in order to provide high capacity storage to host the markup documents along with the instructions to launch and enhance the functionality of the interactive contents via the applets.

**As to independent claim 10,** Stone teaches

processing a markup document classifying tagged applets into bound and unbound applets to display interactive contents (fig. 6; col. 7, lines 30-48; col. 8, lines 67, col. 9, lines 1-6), determining whether an applet execution of the markup document is a bound applet or an unbound applet according to the classifying (fig. 6; col. 7, lines 30-48), and if the applet is an unbound applet, launching the unbound applet by immediately issuing predetermined commands to first set the unbound applet into an initiate state and then into a start state, respectively (col. 7, lines 45-58; col. 8 lines 64-67; col. 9, lines 1-2, 7-14, that by running in a command shell environment, the commands will implement the code to initiate and/or start the applet).

Art Unit: 2179

Stone does not teach an interactive digital versatile disc (DVD) player, comprising: a programmed computer processor controlling the player according to a process.

However, Engstrom teaches an interactive digital versatile disc (DVD) player (par [0054]-[0066]), comprising: a programmed computer processor controlling the player according to a process (par [0054]-[0066]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Stone by having an interactive digital versatile disc (DVD) player, comprising: a programmed computer processor controlling the player according to a process as taught by Engstrom in order to provide high capacity program computer processor to launch the markup document providing the interactive content in which the applets can be bound or unbound to the markup document.

**As to dependent claim 11**, Stone further teaches the programmed computer processor launches the unbound applet without synchronization with the markup document processing (col. 7, lines 34-38).

**As to dependent claim 12**, Stone further teaches the launched unbound applet continuously executes independent of the markup document processing (fig. 6, label 602, col. 7, lines 34-36).

*It is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332- 33,216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006,1009, 158 USPQ 275, 277 (CCPA 1968)).*

*The Examiner notes MPEP § 2144.01, that quotes In re Preda, 401 F.2d 825,159 USPQ 342, 344 (CCPA 1968) as stating "in considering the disclosure of a*

Art Unit: 2179

*reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." Further MPEP 2123, states that "a reference may be relied upon for all that it would have reasonably suggested to one having ordinary skill the art, including nonpreferred embodiments. Merck & Co. v. Biocraft Laboratories, 874 F.2d 804, 10 USPQ2d 1843 (Fed. Cir.), cert. denied, 493 U.S. 975 (1989).*

### ***Conclusion***

6. The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. 1.111(c) to consider these references fully when responding to this action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thuy Osberg whose telephone number is 571-270-1258. The examiner can normally be reached on Monday-Friday (8:30AM-5:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO

Art Unit: 2179

Customer Service Representative or access to the automated information system, call

800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TTO

BA HUYNH  
PRIMARY EXAMINER